Raha Imanirad, Xin-She Yang, Julian Scott Yeomans / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 3, Issue 2, March -April 2013, pp.1677-1686

A Biologically-Inspired Metaheuristic Procedure for Modelling-to-Generate-Alternatives

Raha Imanirad*, Xin-She Yang ** and Julian Scott Yeomans***

* (OMIS Area, Schulich School of Business, York University,4700 Keele Street, Toronto, ON, M3J 1P3 Canada)

** (Department of Design Engineering and Mathematics,School of Science and Technology, Middlesex UniversityHendon Campus, London NW4 4BT, UK)

*** (OMIS Area, Schulich School of Business, York University,4700 Keele Street, Toronto, ON, M3J 1P3 Canada)

Abstract

In finding solutions to many "real world" engineering optimizationproblems, it is generally desirable to be able to construct several quantifiably good alternatives that provide very different perspectives to the particular problem. This is because complex decision-making situations typically involve problems riddled with incompatible performance objectives and possess competing design requirements that are very difficult – if not impossible – to quantify and capture when the supporting decision models must be formulated. There are invariably unmodelled design issues, not apparent during model construction, which can greatly impact the acceptability of any model's solutions. Consequently, it is preferable to generate numerous alternatives that provide dissimilar approaches to the problem. These alternatives should possess near-optimal objective measures with respect to all known modelled objective(s), but be fundamentally different from each other in terms of the system structures characterized by their decision variables. This maximally different solution creation approach is referred to as modelling-to-generate-alternatives (MGA). This paper provides an efficient, biologically-inspired metaheuristic MGA method that can concurrently create multiple solution alternatives that simultaneously satisfy the required system performance criteria and are maximally different in the decision space. The efficacy of this MGA approach is demonstrated on a number of benchmark engineering optimizationproblems.

Keywords-Biologically-inspired Metaheuristic Algorithms,Firefly Algorithm, Modelling-to-generate-alternatives

1. Introduction

Typical "real world" engineering decision-making involves complex problems that possess design requirements which are frequently very difficult to incorporate into their supporting mathematical programming formulations and tend to be plagued by numerous unquantifiable components [1-3]. While mathematically optimal solutions may provide the best answers to these modelled formulations, they are generally not the best solutions to the underlying real problems as there are invariably unmodelled aspects not apparent during the model construction phase [1-2,4-6]. Hence, it is generally considered desirable to generate a reasonable number of very different alternatives that provide multiple, contrasting perspectives to the specified problem [4,7-8]. These alternatives should preferably all possess near-optimal objective measures with respect to all of the modelled objective(s), but be as different as possible from each other in terms of the system structures characterized by their decision variables. Several approaches collectively referred to as *modelling-to-generate-alternatives* (MGA) have been developed in response to this multi-solution creation requirement [6,8-9].

The primary motivation behind MGA is to construct a manageably small set of alternatives that are good with respect to all measured objective(s) yet are maximally different from each other within the prescribed decision space [6,9]. The resulting set of alternatives should provide diverse approaches that all perform similarly with respect to the known modelled objectives, yet very differently with respect to any unmodelled issues [3]. Obviously the decision-makers must subsequently conduct a comprehensive comparison of these alternatives to determine which option(s) would most closely satisfy their very specific circumstances. Consequently, MGA approaches are necessarily considered as decision support processes rather than the role of explicit solution determination methods assumed, in general, for optimization.

In this paper, it is shown how to efficiently generate sets of maximally different solution alternatives by employing a modified version of the biologically-inspired Firefly Algorithm (FA) of Yang [10-11] combined with a concurrent, co-evolutionary MGA approach [12-13]. For calculation and optimization purposes, Yang [11] has demonstrated that the FA is more computationally efficient than such commonly-employed metaheuristic procedures as genetic algorithms, simulated annealing, and enhanced particle swarm optimization [14-15]. The new FA-driven MGA procedure exploits the earlier approach of Imanirad *et al.* [12-13] by extending the concept of co-evolution into the FA's solution approach to concurrently generate the desired number of solution alternatives. Remarkably, this novel algorithm simultaneously produces the global optimal solution together with *n* locally optimal, maximally different alternatives in a single computational run. Hence, this concurrent co-evolutionary FA-driven procedure is extremely computationally efficient for MGA purposes. The efficacy of this approach for constructing multiple, maximally different solution alternatives for engineering optimization is illustrated using a highly non-linear optimization problem [6], a benchmark multimodal constrained optimization problem [16] and a well-known engineering optimization test problem [14].

2. Modelling To Generate Alternatives

Most mathematical programming methods appearing in the optimization literature have concentrated almost exclusively upon producing solitary optimal solutions to single-objective problem instances or, equivalently, generating noninferior solution sets to multi-objective formulations [2-3,9]. While such algorithms may efficiently generate solutions to the derived complex mathematical models, whether these outputs actually establish "best" approaches to the underlying real problems is certainly questionable [1-2,6,9]. In most "real world" decision environments, there are innumerable system objectives and requirements that are never explicitly apparent or included in the decision formulation stage [1,3-5,7-8]. Furthermore, it may never be possible to explicitly express all of the subjective components because there are frequently numerous incompatible, competing, design requirements and, perhaps, adversarial stakeholder groups involved [4-5]. Therefore most subjective aspects of a problem necessarily remain unquantified and unmodelled in the resultant decision models [7]. This is a common occurrence in situations where final decisions are constructed based not only upon clearly stated and modelled objectives, but also upon more fundamentally subjective socio-political-economic goals and stakeholder preferences [8]. Numerous "real world" examples describing these types of incongruent modelling dualities appear in Loughlin *et al.* [6], Brill *et al.* [9], Baugh *et al.* [17] and Zechman & Ranjithan [18].

When unquantified issues and unmodelled objectives exist, non-conventional approaches are required that not only search the decision space for noninferior sets of solutions, but must also explore the decision space for discernibly *inferior* alternatives to the modelled problem. In particular, any search for good alternatives to problems known or suspected to contain unmodelled objectives must focus not only on the non-inferior solution set, but also necessarily on an explicit exploration of the problem's inferior region.

To illustrate the implications of an unmodelled objective on a decision search, assume that the optimal solution for a quantified, single-objective, maximization decision problem is $\boldsymbol{X}^*$ with corresponding objective value $Z1^*$. Now suppose that there exists a second, unmodelled, maximization objective $Z2$ that subjectively reflects some unquantifiable "political acceptability" component. Let the solution $\boldsymbol{X}^a$, belonging to the noninferior, 2-objective set, represent a potential best compromise solution if both objectives could somehow have been simultaneously evaluated by the decision-maker. While $\boldsymbol{X}^a$ might be viewed as the best compromise solution to the real problem, it would appear inferior to the solution $\boldsymbol{X}^*$ in the quantified mathematical model, since it must be the case that $Z1^a \leq Z1^*$. Consequently, when unmodelled objectives are factored into the decision making process, mathematically inferior solutions for the modelled problem can prove optimal to the underlying "reallife" problem.

Therefore, when unmodelled objectives and unquantified issues might exist, different solution approaches are needed in order to not only search the decision space for the noninferior set of solutions, but also to simultaneously explore the decision space for inferior alternative solutions to the modelled problem. Population-based solution methods such as the FA permit concurrent searches throughout a feasible region and thus prove to be particularly adept procedures for searching through a problem's decision space.

The primary motivation behind MGA is to produce a manageably small set of alternatives that are quantifiably good with respect to the known modelled objectives yet are as different as possible from each other in the decision space. In doing this, the resulting alternative solution set is likely to provide truly different choices that all perform somewhat similarly with respect to the modelled objective(s) yet very differently with respect to any unknown unmodelled issues. By generating such a set of good-but-different solutions, the decision-makers can explore desirable qualities within the alternatives that may prove to satisfactorily address the various unmodelled objectives to varying degrees of stakeholder acceptability.

In order to properly motivate an MGA search procedure, it is necessary to apply a more mathematically formal definition to the goals of the MGA process [6,8]. Suppose the optimal solution to an original mathematical model is $\boldsymbol{X^*}$ with objective value $\boldsymbol{Z^*} = F(\boldsymbol{X^*})$. The following model can then be solved to generate an alternative solution that is maximally different from $\boldsymbol{X^*}$:

Maximize $\Delta = \sum_i \; | X_i - X_i^* |$ [P1]

Subject to: $\boldsymbol{X} \in D$
$| F(\boldsymbol{X}) - \boldsymbol{Z}^* | \leq$ T

where Δ represents some difference function (for clarity, shown as an absolute difference in this instance), D is the original mathematical model's feasible domain and T is a targeted tolerance value specified relative to the problem's original optimal objective $\boldsymbol{Z}^*$. T is a user-supplied value that determines how much of the inferior region is to be explored in the search for acceptable alternative solutions.

3. Firefly Algorithm For Function Optimization

While this section supplies only a relatively brief synopsis of the FA procedure, more detailed explanations can be accessed in Yang [10-11], Imanirad *et al*. [12-13] and Gandomi *et al.* [15]. The FA is a biologically-inspired, population-based metaheuristic. Each firefly in the population represents one potential solution to a problem and the population of fireflies should initially be distributed uniformly and randomly throughout the solution space. The solution approach employs three idealized rules. (i) The brightness of a firefly is determined by the overall landscape of the objective function. Namely, for a maximization problem, the brightness is simply considered to be proportional to the value of the objective function. (ii) The relative attractiveness between any two fireflies is directly proportional to their respective brightness. This implies that for any two flashing fireflies, the less bright firefly will always be inclined to move towards the brighter one. However, attractiveness and brightness both decrease as the relative distance between the fireflies increases. If there is no brighter firefly within its visible neighborhood, then the particular firefly will move about randomly. (iii) All fireflies within the population are considered unisex, so that any one firefly could potentially be attracted to any other firefly irrespective of their sex. Based upon these three rules, the basic operational steps of the FA can be summarized within the pseudo-code of Fig.1 [11].

Objective Function $F(\boldsymbol{X})$, $\boldsymbol{X} = (x_1, x_2, \ldots x_d)$
Generate the initial population of n fireflies, $\boldsymbol{X}_i$, $i = 1, 2, \ldots, n$
Light intensity I_i at $\boldsymbol{X}_i$ is determined by $F(\boldsymbol{X}_i)$
Define the light absorption coefficient γ
while (t < MaxGeneration)
for i = 1: n , all n fireflies
for j = 1: n ,all n fireflies (inner loop)
  if ($I_i < I_j$), Move firefly i towards j; **end if**
  Vary attractiveness with distance r via $e^{-\gamma r}$
endfor j
end for i
Rank the fireflies and find the current global best solution $\boldsymbol{G}^*$
end while
Postprocess the results

Figure 1: pseudo code of the firefly algorithm

In the FA, there are two important issues to resolve: the formulation of attractiveness and the variation of light intensity. For simplicity, it can always be assumed that the attractiveness of a firefly is determined by its brightness which in turn is associated with its encoded objective function value. In the simplest case, the brightness of a firefly at a particular location $\boldsymbol{X}$ would be its calculated objective value $F(\boldsymbol{X})$. However, the attractiveness, □, between fireflies is relative and will vary with the distance r_{ij} between firefly i and firefly j. In addition, light intensity decreases with the distance from its source, and light is also absorbed in the media, so the attractiveness needs to vary with the degree of absorption. Consequently, the overall attractiveness of a firefly can be defined as:

$\beta = \beta_0 \exp(-\gamma r^2)$

where β_0 is the attractiveness at distance r = 0 and γ is the fixed light absorption coefficient for the specific medium. If the distance r_{ij} between any two fireflies i and j located at $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, respectively, is calculated using the Euclidean norm, then the movement of a firefly i that is attracted to another more attractive (i.e. brighter) firefly j is determined by:

$\boldsymbol{X}_i = \boldsymbol{X}_i + \beta_0 \exp(-\gamma (r_{ij})^2)(\boldsymbol{X}_i - \boldsymbol{X}_j) + \alpha \boldsymbol{\varepsilon}_i$

In this expression of movement, the second term is due to the relative attraction and the third term is a randomization component. Yang [11] indicates that α is a randomization parameter normally selected within the range [0,1] and $\boldsymbol{\varepsilon}_i$ is a vector of random numbers drawn from either a Gaussian or uniform (generally [-0.5,0.5]) distribution. It should be explicitly noted that this expression represents a random walk biased toward brighter fireflies and if β_0 = 0, it becomes a simple random walk. The parameter γ characterizes the variation of the attractiveness and its value determines the speed of the algorithm's convergence. For most applications, γ is typically set between 0.1 to 10 [11,15]. In any given optimization problem, for a very large number of fireflies n>>k, where k is the number of local optima, the initial locations of the n fireflies should be distributed relatively uniformly throughout the entire search space. As the FA proceeds, the fireflies begin to converge into all of the local optima (including the global ones). Hence, by comparing the best solutions among all these optima, the global optima can easily be determined. Yang (2010) proves that the FA will approach the global optima when $n \rightarrow \infty$ and the number of iterations t, is set so that t>>1. In reality, the FA has been found to converge extremely quickly with n set in the range 20 to 50 [10,15].

Two important limiting or asymptotic cases occur when $\gamma \rightarrow 0$ and when $\gamma \rightarrow \infty$. For $\gamma \rightarrow 0$, the attractiveness is constant $\beta = \beta_0$, which is equivalent to having a light intensity that does not decrease. Thus, a firefly would be visible to every other firefly anywhere within the solution domain. Hence, a single (usually global) optima can easily be reached. If the inner loop for j in Fig.1 is removed and $\boldsymbol{X}_j$ is replaced by the current global best $\boldsymbol{G^*}$, then this implies that the FA reverts to a special case of the accelerated particle swarm optimization (PSO) algorithm. Subsequently, the computational efficiency of this special FA case is equivalent to that of enhanced PSO. Conversely, when $\gamma \rightarrow \infty$, the attractiveness is essentially zero along the sightline of all other fireflies. This is equivalent to the case where the fireflies randomly roam throughout a very thick foggy region with no other fireflies visible and each firefly roams in a completely random fashion. This case corresponds to a completely random search method. As the FA operates between these two asymptotic extremes, it is possible to adjust the parameters α and γ so that the FA can outperform both a random search and the enhanced PSO algorithms [15].

The computational efficiencies of the FA will be exploited in the subsequent MGA solution approach. As noted, between the two asymptotic extremes, the population in the FA can determine both the global optima as well as the local optima concurrently. This concurrency of population-based solution procedures holds huge computational and efficiency advantages for MGA purposes [8]. An additional advantage of the FA for MGA implementation is that the different fireflies essentially work independently of each other, implying that FA procedures are better than genetic algorithms and PSO for MGA because the fireflies will tend to aggregate more closely around each local optimum [11,15]. Consequently, with a judicious selection of parameter settings, the FA will simultaneously converge extremely quickly into both local and global optima [10-11,15].

4. FA-Driven Computational Algorithm For Concurrent MGA

The FA-driven MGA approach to be introduced is designed to generate a pre-determined small number of close-to-optimal, but maximally different alternatives, by essentially adjusting the value of T in [P1] and using the FA to solve the corresponding, maximal difference problem instance. By exploiting the co-evolutionary solution structure within the population of the FA, stratified subpopulations within the algorithm's overall population are established as the Fireflies collectively evolve toward different local optima within the solution space. In this process, each desired solution alternative undergoes the common search procedure of the FA. However, the survival of solutions depends both upon how well the solutions perform with respect to both the modelled objective(s) and by how far away they are from all of the other alternatives generated in the decision space.

A direct process for generating alternatives with the FA would be to iteratively solve the maximum difference model [P1] by incrementally updating the target T whenever a new alternative needs to be produced and then re-running the algorithm. This iterative approach would parallel the original Hop, Skip, and Jump (HSJ) MGA algorithm of Brill *et al*. [9] in which, once an initial problem formulation has been optimized, supplementary alternatives are iteratively created through a systematic, incremental adjustment of the target constraint to force the sequential generation of the suboptimal solutions. While this approach is straightforward, it requires a repeated execution of the specific optimization algorithm employed [8,12-13].

In contrast, the concurrent MGA approach is designed to generate the pre-determined number of maximally different alternatives within the entire population in a single run of the FA procedure (i.e. the same number of runs as if FA were used solely for function optimization purposes) and its efficiency is based upon the concept of co-evolution [12-13]. In this FA-driven co-evolutionary approach, pre-specified stratified subpopulation ranges within the FA's overall population are established that collectively evolve the search toward the creation of the stipulated number of maximally different alternatives. Each desired solution alternative is represented by each respective subpopulation and each subpopulation undergoes the common processing operations of the FA.

The FA-driven approach can be structured upon any standard FA solution procedure containing appropriate encodings and operators that best correspond to the problem. The survival of solutions in each subpopulation depends simultaneously upon how well the solutions perform with respect to the modelled objective(s) and by how far away they are from all of the other alternatives. Consequently, the evolution of solutions in each subpopulation toward local optima is directly influenced by those solutions currently existing in all of the other subpopulations, which necessarily forces the concurrent co-evolution of each subpopulation towards good but maximally distant regions of the decision space. This co-evolutionary concept enables the simultaneous search for, and production of, the set of quantifiably good solutions that are maximally different from each other according to [P1] [8].

By employing this co-evolutionary concept, it becomes possible to implement an FA-driven

MGA procedure that concurrently produces alternatives which possess objective function bounds that are somewhat analogous, but inherently superior, to those created by the sequential, iterative HSJ-styled solution generation approach. While each alternative produced by an HSJ procedure is maximally different only from the single, overall optimal solution together with a bound on the objective value which is at least x% different from the best objective (i.e. x = 1%, 2%, etc.), the concurrent co-evolutionary procedure is able to generate alternatives that are no more than x% different from the overall optimal solution but with each one of these solutions being as maximally different as possible from every other generated alternative that is produced. Co-evolution is also much more efficient than the sequential HSJ-styled approach in that it exploits the inherent population-based searches of FA procedures to concurrently generate the entire set of maximally different solutions using only a single population. Namely, while an HSJ-styled approach would need to run n different times in order to generate n different alternatives, the concurrent algorithm is required to run only once to produce its entire set of maximally different alternatives irrespective of the value of n. Hence, it is a much more computationally efficient solution generation process.

The steps in the FA-driven co-evolutionary MGA algorithm are as follows:

1. Create the initial population stratified into P equally-sized subpopulations. P represents the desired number of maximally different alternative solutions within a prescribed target deviation from the optimal to be generated. S_p represents the p^{th} subpopulation set of solutions, $p = 1,\ldots,P$ and there are K solutions contained within each S_p. Note: The value for P must be set *a priori* by the decision-maker.
2. Evaluate all solutions in S_p, $p = 1,\ldots,P$, with respect to the modelled objective. Solutions meeting the target constraint and all other problem constraints are designated as *feasible*, while all other solutions are designated as *infeasible*.
3. Apply an appropriate elitism operator to each S_p to preserve the best individual in each subpopulation. In S_p, $p = 1,\ldots,P$, the best solution is the feasible solution most distant in decision space from all of the other subpopulations (the distance measure is defined in Step 6). Note: Because the best solution to date is always placed into each subpopulation, at least one solution in S_p will always be feasible. This step simultaneously selects a set of alternatives that respectively satisfy different values of the target T while being as far apart as possible (i.e. maximally different in the sense of [P1]) from the solutions generated in each of the other subpopulations. By the co-evolutionary nature of this algorithm, the alternatives are simultaneously generated in one pass of the procedure rather than the P implementations suggested by the necessary increments to T in problem [P1].
4. Stop the algorithm if the termination criteria (such as maximum number of iterations or some measure of solution convergence) are met. Otherwise, proceed to Step 5.
5. Identify the decision space centroid, C_{ip}, for each of the $K' \leq K$ feasible solutions within $k = 1,\ldots,K$ of S_p, for each of the N decision variables X_{ikp}, $i = 1,\ldots, N$. Each centroid represents the N-dimensional centre of mass for the solutions in each of the respective subpopulations, p. As an illustrative example for determining a centroid, calculate $C_{ip} = (1/K')* \sum_k X_{ikp}$. In this calculation, each dimension of each centroid is computed as the straightforward average value of that decision variable over all of the values for that variable within the feasible solutions of the respective subpopulation. Alternatively, a centroid could be calculated as some fitness-weighted average or by some other appropriately defined measure.
6. For each solution $k = 1,\ldots, K$, in each S_q, calculate D_{kq}, a distance measure between that solution and all other subpopulations. As an illustrative example for determining a distance measure, calculate $D_{kq} = \text{Min} \{\sum_i |X_{ikp} - C_{ip}|;\ p = 1,\ldots,P,\ p \neq q\}$. This distance represents the minimum distance between solution k in subpopulation q and the centroids of all other subpopulations. Alternatively, the distance measure could be calculated by some other appropriately defined function.
7. Rank the solutions within each S_p according to the distance measure D_{kq} objective – appropriately adjusted to incorporate any constraint violation penalties. The goal of maximal difference is to force solutions from one subpopulation to be as far apart as possible in the decision space from the solutions of each of the other subpopulations. This step orders the specific solutions in each subpopulation by those solutions which are most distant from the solutions in all of the other subpopulations.
8. In each S_p, apply the appropriate FA "change operations" to the solutions and return to Step 2.

5. Computational Testing Of The Firefly Algorithm Used For MGA

As described in the previous sections, "real world" decision-makers generally prefer to be able to select from a set of "near-optimal" alternatives that significantly differ from each other in terms of the system structures characterized by their decision variables. The ability of the co-evolutionary FA-driven MGA procedure to concurrently produce such maximally different alternatives will be demonstrated using a 100-peak multimodal optimization problem taken from Loughlin *et al.* [6], a benchmark non-linear constrained optimization problem from Aragon *et al.* [16] and a widely-tested constrained engineering optimization problem from Cagnina *et al*. [14].

The mathematical formulation for the multimodal test problem of Loughlin *et al*. [6] is:

$$\text{Maximize } F(x, y) = \text{Sin}(19\pi x) + \frac{x}{1.7} + \text{Sin}(19\pi y) + \frac{y}{1.7} + 2$$

$$0.0 \leq x \leq 1.0$$
$$0.0 \leq y \leq 1.0$$

The feasible region corresponding to this problem contains 100 peaks separated by valleys with the increasing amplitudes for both the peaks and valleys as the values of the decision variables increase from their lower bounds of (0,0) toward their upper limits at (1,1). For the design parameters employed in this specific problem formulation, the mathematically optimal solution of $F(x, y) = 5.146$ occurs at point $(x, y) = (0.974, 0.974)$ [6].

As described in the previous section, in order to create the desired set of maximally different alternatives for this problem, it would be possible to insert extra target constraints in an incrementally increasing fashion into the original mathematical formulation to force the generation of solutions that were structurally different from the initial optimal solution. Suppose for example that ten additional solution options were to be created through the inclusion of a technical constraint that increased value of the objective in the original model formulation from 1% up to 10% in increments of 1%. By adding these incremental target constraints to the original model and sequentially resolving the problem 10 times, it would be possible to create the prescribed number of maximally different alternatives.

Table 1. Objective Values and Solutions for the 11 Maximally Different Alternatives

Increment	**1% Increment Between Alternatives**			**2.5% Increment Between Alternatives**		
	***F*(*x*,*y*)**	*x*	*Y*	***F*(*x*,*y*)**	*X*	*Y*
Optimal	5.14	0.97	0.97	5.14	0.97	0.97
Alternative 1	5.10	0.98	0.97	5.01	0.87	0.87
Alternative 2	5.05	0.87	0.98	4.89	0.66	0.87
Alternative 3	5.00	0.76	0.98	4.77	0.87	0.45
Alternative 4	4.99	0.98	0.87	4.65	0.33	0.97
Alternative 5	4.91	0.98	0.76	4.50	0.98	0.02
Alternative 6	4.89	0.55	0.97	4.43	0.02	0.98
Alternative 7	4.89	0.98	0.55	4.29	0.98	0.02
Alternative 8	4.74	0.34	0.98	4.13	0.02	0.99
Alternative 9	4.69	0.98	0.24	4.02	0.99	0.02
Alternative 10	4.64	0.13	0.98	3.87	0.01	0.98

However, to improve upon the process of running ten separate additional instances, the co-evolutionary FA MGA method could be run exactly once to concurrently produce all of the desired alternatives. By employing the co-evolutionary FA-driven MGA algorithm from the previous section, the optimal solution together with 10 maximally different solutions to it were generated for alternative increments of 1% and 2.5%, respectively (see Table 1).

The second alternative generation example involves the constrained non-linear optimization test problem of Aragon *et al.* [16]. The mathematical formulation for this problem is:

$$\text{Min } F(\mathbf{X}) = (x_1-10)^2+5(x_2-12)^2+x_3^4+3(x_4-11)^2+10x_5^6+7x_6^2+x_7^4-4x_6x_7-10x_6-8x_7$$

Subject to:

$$g_1(\mathbf{X}) = 2x_1^2+3x_2^4+x_3+4x_4^2+5x_5-127 \le 0$$

$$g_2(\mathbf{X}) = 7x_1+3x_2+10x_3^2+x_4-x_5-282 \le 0$$

$$g_3(\mathbf{X}) = 23x_1+x_2^2+6x_6^2-8x_7-196 \le 0$$

$$g_4(\mathbf{X}) = 4x_1^2+x_2^2-3x_1x_2+2x_3^2+5x_6-12x_7 \le 0$$

$$-10 \le x_i \le 10, \quad i = 1, 2, 3, 4, 5, 6, 7$$

Table 2. Objective Values and Solutions for the 11 Maximally Different Alternatives

Increment	1% Increment Between Alternatives							
	F(X)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Optimal	680.630	2.3304	1.9513	-0.4775	4.3657	0.6244	1.0381	1.5942
Alternative 1	683.917	2.3025	1.9353	-0.4881	4.3333	-0.6169	1.0355	1.5889
Alternative 2	687.580	2.2892	1.8985	-0.4605	4.3364	-0.5962	1.0208	1.5782
Alternative 3	696.899	2.2934	1.9096	-0.4397	4.3369	-0.6616	1.0331	1.6176
Alternative 4	705.926	2.3080	1.9171	-0.4724	4.3343	-0.6578	1.053	1.6078
Alternative 5	706.837	2.2913	1.9003	-0.3965	4.3548	-0.6388	1.0796	1.6023
Alternative 6	718.478	2.2904	1.9037	-0.427	4.3637	-0.5871	0.9955	1.6230
Alternative 7	725.652	2.3428	1.9158	-0.4459	4.3929	-0.6672	1.0382	1.6129
Alternative 8	730.091	2.2892	1.8985	-0.4605	4.3364	-0.5962	1.0208	1.5782
Alternative 9	741.897	2.2904	1.9037	-0.427	4.3637	-0.5871	0.9955	1.6230
Alternative 10	747.925	2.3577	1.9121	-0.4395	4.3314	-0.5869	1.0038	1.6148

Table 3. Objective Values and Solutions for the 11 Maximally Different Alternatives

Increment	2.5% Increment Between Alternatives							
	F(X)	x_1	x_2	x_3	x_4	x_5	x_6	x_7
Optimal	680.630	2.3304	1.9513	-0.4775	4.3657	0.6244	1.0381	1.5942
Alternative 1	687.022	2.3056	1.9076	-0.4245	4.3256	-0.6184	1.0388	1.6067
Alternative 2	711.793	2.3174	1.9111	-0.4084	4.3668	-0.6166	1.0759	1.6116
Alternative 3	730.671	2.2916	1.9496	-0.4442	4.3474	-0.6154	1.0555	1.5864
Alternative 4	744.901	2.3468	1.9118	-0.4087	4.3557	-0.6283	0.9899	1.6024
Alternative 5	756.260	2.2985	1.9019	-0.4452	4.3577	-0.5927	1.0022	1.5770
Alternative 6	779.735	2.3463	1.9397	-0.4338	4.3425	-0.5867	1.0457	1.6301
Alternative 7	796.641	2.3011	1.9128	-0.4282	4.3386	-0.5758	1.0035	1.6227
Alternative 8	811.767	2.3539	1.9579	-0.4543	4.3338	-0.6425	1.0413	1.6155
Alternative 9	832.123	2.3690	1.9208	-0.4181	4.4001	-0.617	1.0411	1.6374
Alternative 10	846.019	2.2967	1.897	-0.4684	4.3467	-0.6374	1.0252	1.5721

The optimal solution for the specific design parameters employed within this formulation is F(***X****) = 680.6300573 with decision variable values of ***X**** = (2.330499, 1.951372, -0.4775414, 4.365726, 0.6244870, 1.038131, 1.594227) [16]. The FA-driven MGA algorithm was run exactly once to generate the optimal solution and the 10 maximally different solutions shown in Tables 2 and 3.

The third illustration will apply the FA-driven MGA procedure to the spring design problem taken from Cagnina *et al.* [14]. The design of a tension and compression spring has frequently been employed as a standard benchmark test problem for constrained engineering optimization algorithms [14]. The problem involves three design variables: (i) x_1, the wire diameter, (ii) x_2, the coil diameter, and (iii) x_3, the length of the coil. The aim is to essentially minimize the weight subject to constraints on deflection, stress, surge frequency and geometry. The mathematical formulation for this test problem can be summarized as:

Minimize $F(\boldsymbol{X}) = x_1^2 x_2 (2 + x_3)$

Subject to:

$$g_1(\boldsymbol{X}) = 1 - \frac{x_2^3 x_3}{71785 x_1^4} \le 0$$

$$g_2(\boldsymbol{X}) = \frac{4x_2^2 - x_1 x_2}{12566(x_1^3 x_2 - x_1^4)} + \frac{1}{5108\, x_1^2} - 1 \le 0$$

$$g_3(\boldsymbol{X}) = 1 - \frac{140.45 x_1}{x_2^2 x_3} \le 0$$

$$g_4(\boldsymbol{X}) = \frac{x_1 + x_2}{1.5} - 1 \le 0$$

$0.05 \le x_1 \le 2.00.25 \le x_2 \le 1.3 \;\; 2.0 \le x_3 \le 15.0$

The optimal solution for the specific design parameters employed within this formulation is $F(\boldsymbol{X^*}) = 0.0127$ with decision variable values of $\boldsymbol{X^*} = (0.051690, 0.356750, 11.287126)$ [14]. The MGA procedure was used to create the optimal solution and the 10 maximally different solutionsshown in Table 4.

Table 4. Objective Values and Solutions for the 11 Maximally Different Alternatives

Increment	**1% Increment Between Alternatives**				**2.5% Increment Between Alternatives**			
	F(*X*)	x_1	x_2	x_3	**F(*X*)**	x_1	x_2	x_3
Optimal	0.0127	0.0517	0.3567	11.2871	0.0127	0.0517	0.3567	11.2871
Alternative 1	0.0128	0.05	0.3164	14.1754	0.0128	0.05	0.3165	14.1598
Alternative 2	0.0128	0.0514	0.3472	12.0089	0.0131	0.05	0.3129	14.777
Alternative 3	0.0129	0.0529	0.3862	9.9684	0.0132	0.05	0.3167	14.6402
Alternative 4	0.013	0.0521	0.3656	11.0667	0.0140	0.0557	0.4307	9.5783
Alternative 5	0.0131	0.0527	0.3766	10.5179	0.0143	0.0542	0.4014	11.6481
Alternative 6	0.0134	0.05	0.3157	14.978	0.0146	0.0546	0.4247	10.7556
Alternative 7	0.0135	0.0524	0.3597	11.6966	0.0149	0.0562	0.438	11.1197
Alternative 8	0.0137	0.052	0.3629	12.1615	0.0152	0.0605	0.4836	8.9963
Alternative 9	0.0138	0.0523	0.348	13.3247	0.0156	0.0574	0.3841	14.5182
Alternative 10	0.0140	0.0535	0.3857	14.162	0.0159	0.0553	0.4072	15.0000

These computational examples underscore several important findings with respect to the concurrent FA-driven MGA method: (i) The co-evolutionary capabilities within the FA can be exploited to generate more good alternatives than planners would be able to create using other MGA approaches because of the evolving nature of its population-based solution searches; (ii) By the design of the MGA algorithm, the alternatives generated are good for planning purposes since all of their structures will be as mutually and maximally different from each other as possible (i.e. these differences are not just simply different from the overall optimal solution as in the HSJ-style approach to MGA); (iii) The approach is very computationally efficient since it need only be run once to generate its entire set of multiple, good solution alternatives (i.e. to generate *n* solution alternatives, the MGA algorithm needs to run exactly the same number of times that the FA would need to be run for function optimization purposes alone – namely once – irrespective of the value of *n*); and, (iv) The best overall solutions produced by the MGA procedure will be very similar, if not identical, to the best overall solutions that would be produced by the FA for function optimization alone.

In summary, the three examples in this section have demonstrated how the MGA modelling perspective can be used to concurrently generate multiple alternatives that satisfy known system performance criteria according to the prespecified bounds and yet remain as maximally different from each other as possible in the decision space. In addition to its alternative generating capabilities, the FA component within the MGA approach simultaneously performs extremely well with respect to its role in function optimization. It should be explicitly noted that the overall best solutions produced by the FA-driven MGA procedure for the test problems are indistinguishable from the optimal ones determined by Loughlin *et al.* [6], Aragon *et al.* [16] and Cagnina *et al.* [14].

6. Conclusions

"Real world" engineering optimization problems generally possess multidimensional performance specifications that are compounded by incompatible performance objectives and unquantifiable modelling features. These problems usually contain incongruent design requirements which are very difficult – if not impossible – to capture at the time that supporting decision models are formulated. Consequently, there are invariably unmodelled problem facets, not apparent during the model construction, that can greatly impact the acceptability of the model's solutions. These uncertain and competing dimensions force decision-makers to integrate many conflicting sources into their decision process prior to final solution construction. Faced with such incongruencies, it is unlikely that any single solution could ever be constructed that simultaneously satisfies all of the ambiguous system requirements.

Therefore, any ancillary modelling techniques used to support decision formulation have to somehow simultaneously account for all of these features while being flexible enough to encapsulate the impacts from the inherent planning uncertainties. Under such circumstances, it is preferable to create a set of quantifiably good alternatives that provide very different perspectives to the potentially unmodelled performance design issues during the problem formulation stage. The unique performance features captured within these dissimilar alternatives can result in very different system performance with respect to the unmodelled issues, hopefully thereby addressing some of the unmodelled issues into the actual solution process.

In this paper, an FA-driven MGA approach was introduced that demonstrated how the computationally efficient FA could be exploited to concurrently generate multiple, maximally different, near-best alternatives via the co-evolutionary nature of its population-based solution technique. In this MGA capacity, the algorithm produces numerous solutions possessing the requisite problem characteristics, with each generated alternative guaranteeing a very different perspective. Since FA-driven techniques can be adapted to solve a wide variety of problem types, the practicality of this MGA approach can clearly be extended into numerous disparate "real world" applications. These extensions will become the focus of future research.

References

[1] Brugnach, M., Tagg, A., Keil, F., and De Lange W.J., Uncertainty matters: computer models at the science-policy interface, *Water Resources Management21*, 2007, 1075-1090.

[2] Janssen, J.A.E.B., Krol, M.S., Schielen, R.M.J., and Hoekstra, A.Y., The effect of modelling quantified expert knowledge and uncertainty information on model based decision making,*Environmental Science and Policy13(3)*,2010, 229-238.

[3] Walker, W.E., Harremoes, P., Rotmans, J., Van der Sluis, J.P., Van Asselt, M.B.A., Janssen, P., and Krayer von Krauss, M.P., Defining uncertainty – a conceptual basis for uncertainty management in model-based decision support,*Integrated Assessment4(1)*,2003, 5-17.

[4] Gunalay, Y., and Yeomans, J. S., Generating alternatives using simulationoptimization combined with niching operators to address unmodelled objectives in a waste management facility expansion planning case, *International Journal of Operations Research and Information Systems*, 2012, In Press.

[5] Gunalay, Y., Yeomans, J. S., and Huang, G. H., Modelling to generate alternative policies in highly uncertain environments: An application to municipal solid waste management planning,*Journal of Environmental Informatics*, *19(2)*,2012, 58-69.

[6] Loughlin, D.H., Ranjithan, S.R., Brill, E.D., and Baugh, J.W., Genetic Algorithm Approaches for Addressing Unmodeled Objectives in Optimization Problems,*Engineering Optimization33(5)*,2001, 549-569.

[7] Matthies, M., Giupponi, C., and Ostendorf, B., Environmental decision support systems: Current issues, methods and tools,*Environmental Modelling and Software22(2)*,2007, 123-127.

[8] Yeomans, J.S., and Gunalay, Y., Simulation-Optimization Techniques for Modelling to Generate Alternatives in Waste Management Planning,*Journal of Applied Operational Research3(1)*,2011, 23-35.

[9] Brill, E.D., Chang, S.Y., and Hopkins L.D., Modelling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning, *Management Science28(3)*,1982, 221-235.

[10] Yang, X.S., Firefly Algorithms for Multimodal Optimization,*Lecture Notes in Computer Science5792*, 2009, 169-178.

[11] Yang, X.S., *Nature-Inspired metaheuristic algorithms* 2nd Ed (UK: Luniver Press, Frome, 2010).

[12] Imanirad, R., Yang, X.S., and Yeomans, J.S., A Computationally Efficient, Biologically-Inspired Modelling-to-Generate-Alternatives Method,*Journal on Computing2(2)*, 2012a, 43-47.

[13] Imanirad, R., Yang, X.S., and Yeomans, J.S., A Co-evolutionary, Nature-Inspired Algorithm for the Concurrent Generation of Alternatives,*Journal on Computing2(3)*,2012b ,101-106.

[14] Cagnina, L.C., Esquivel, C.A., and Coello, C.A., Solving Engineering Optimization Problems with the Simple Constrained Particle Swarm Optimizer,*Informatica32*, 2008,319-326.

[15] Gandomi, A.H., Yang X.S., and Alavi,

A.H., Mixed Variable Structural Optimization Using Firefly Algorithm,*Computers and Structures*, *89(23-24)*,2011, 2325-2336.

[16] Aragon, V.S.,Esquivel, S.C., and Coello, C.C.A., A Modified Version of a T-Cell Algorithm for Constrained Optimization Problems,*International Journal for Numerical Methods In Engineering84*, 2010, 51–378.

[17] Baugh, J.W., Caldwell, S.C., and Brill E.D., A Mathematical Programming Approach for Generating Alternatives in Discrete Structural Optimization,*Engineering Optimization28(1)*,1997, 1-31.

[18] Zechman, E.M., and Ranjithan, S.R., An Evolutionary Algorithm to Generate Alternatives (EAGA) for Engineering Optimization Problems,*Engineering Optimization36(5)*,2004, 539-553.